Blink Detection Using OpenCV

# Objective

One of the most active subjects in computer vision is the concept of facial recognition. One element present in many real-time facial recognition algorithms is the ability to detect a human blinking. Blink detection is arguably one of the more robust means of detecting the face, and also has several useful applications in itself. For instance, one could imagine a camera monitoring a sleepy driver. If the algorithm in the camera software detects the driver beginning to blink more slowly, perhaps the driver can be warned that he or she should pull over. The objective of this project is to implement such an algorithm using the OpenCV library.

# Specification

1. Part 1 –Algorithm Proof of Concept (Winter Quarter)
   1. Sotware Development Platform
      1. Windows 7 OS
      2. Microsoft Visual Studio 2010
      3. OpenCV 2.2
   2. Static Image Testing
      1. The system will need to correctly identify whether or not the subject in an image is blinking. The algorithm will output a TRUE or FALSE status based on whether or not the eyes are closed (TRUE indicates closed eyes).
      2. Partially-closed eyes will not be considered a blink. Only completely-closed eyes should be result in TRUE.
      3. Test Images
         1. The algorithm must output the correct result for at least 95% of the test images.
         2. The test images may be composed of both color and greyscale.
         3. Hopefully a large testing image database with blinking subjects can be found (potentially the Color FERET database <http://face.nist.gov/colorferet/>).
         4. If no database can be found, manual images may need to be taken.
   3. Video Testing (time permitting)
      1. If there is enough time in the quarter, the next step would be to implement an engine to run the algorithm in real-time. Perhaps connecting to a web cam would be an easy way to achieve this.
      2. Note that the achieved frame rate may be too slow to detect a blink since most people blink very quickly. If this is true, it will only support the need to move to hardware (Part 2).
      3. If time runs out, this part can probably be moved to Part 2 of the project as a necessary first step (see “Part 2” details below).
2. Part 2 – Hardware Integration (Spring Quarter)
   1. Details of this part of the project are TBD
   2. Tentative plan is to build the source code against the Android NDK so that the algorithm can run on an Android-based mobile smart-phone.